

# Towards Agent-based Crowd Simulation in Airports using Games Technology

Olivier Szymanczyk, Patrick Dickinson, and Tom Duckett

Lincoln School of Computer Science, University of Lincoln. LN6 7TS, United Kingdom  
{oszymanczyk, pdickinson, tduckett}@lincoln.ac.uk

**Abstract.** We adapt popular video games technology for an agent-based crowd simulation in an airport terminal. To achieve this, we investigate the unique traits of airports and implement a virtual crowd by exploiting a scalable layered intelligence technique in combination with physics middleware and a social-forces approach. Our experiments show that the framework runs at interactive frame-rate and evaluate the scalability with increasing number of agents demonstrating navigation behaviour.

**Keywords:** multi agent modelling, pedestrian crowd simulation, game technology, game AI, physics middleware, airport simulation

## 1 Introduction.

The interactions, exchanges and movements of large numbers of pedestrians often lead to complex social and cultural patterns that have been studied since as early as the nineteenth century [8]. Computer scientists are using the recent rise of available computing resources to simulate such crowds. The modelling of crowds has been a significant field of research and many techniques have been developed mainly because of their utility in architecture, safety and security, computer graphics, sociology and entertainment. Despite significant progress in areas such as crowd behaviour, animation and rendering, there are still many fields in which crowd simulation requires additional research. For instance, recent video games have been criticised for feeling lifeless, and researchers have suggested to improve the playing experiences by incorporating believable virtual crowds [19].

Usually crowd simulations involve a trade-off between the complexity of each individual and the size of the crowd. Games AI developers have defined a framework in which agent intelligence is distributed throughout the terrain [21] using a spatial database approach as a solution to this trade-off. The spatial database approach was adapted into a crowd simulation application [1]. Recently games designers have started to use computationally inexpensive middleware physics engines to simulate accurate motion and reaction of objects [13]. A physics-middleware-based crowd simulation was used in our previous work [19].

This work is based on the modelling of a virtual crowd to simulate the London Gatwick Airport South Terminal using games technology. An overview layout of the airport terminal is shown in Fig. 1.



**Fig. 1.** London Gatwick Airport South Terminal Map. Adapted from [11].

Our aim is to design and implement a video game environment that combines a physics middleware engine [13] and an extended scalable layered intelligence technique [21], using Dijkstra's algorithm [5] to pre-compute navigation information, with social-forces driven agents [6] as characters in order to reproduce some unique and interesting airport scenarios, which would in return enrich the feelings of realism and believability for the player in a video game. Although the use of a spatial database and a physics middleware for a social-force-driven crowd simulation is not new [1, 19], to the best of our knowledge, no existing crowd simulation has specifically exploited the uniqueness of airport traveller experience or used the layered AI approach in combination with a physics middleware and the social-forces approach as a backdrop for a real-time virtual crowd setting in a video game situation. We use the available games technology, and extend them to fulfil our aims and goals.

We first discuss the unique traits of airport. Secondly we present our current framework, which is based on the spatial database approach and extend it by augmenting the spread of sociological and environmental data in each cell. This data is consequently used to compute the navigational force acting on each agent. Lastly, we evaluate the computational feasibility of our approach and we discuss its limitations and possible future work.

## **2 Related work**

### **2.1 Crowd simulation**

Several methods that face the challenges of real-time crowd simulation have been proposed [1, 7, 17, 18]. One of the earliest applications of agent-based modelling was seen in Reynold's boid method [17], which has recently been extended onto the Sony Playstation 3 architecture to simulate up to 15,000 agents at 60 frames-per-second [18]. It was concluded that hardware and collision avoidance optimisations were required to enhance the quality of the simulation. Helbing's Social Force Model [6] describes pedestrian dynamics as a force model that takes into account all environmental and sociological forces that act upon each agent. Related work [15] implements this approach in a crowd simulation while extending the model with biologically inspired emotions. However, some undesirable features (unnatural shaking, overlapping and vibrating) were reported. A Cellular Automaton (CA) has been described as an effective technique for modelling complex emergent behaviour of crowds, which has been used to simulate evacuation systems [20]. Despite CA being described as simple and fast to implement [20], the major disadvantage is the inability of individuals to come in direct contact with other individuals. CA crowds have been criticised for providing unrealistic crowd animation [15]. Bannerjee et al. [1] uses a scalable layered intelligence technique and a Semi-Markov Decision Process for a crowd simulation. They demonstrate navigation behaviours in a hall-evacuation scenario and they evaluate the scalability and computational performance of their approach in highly crowded scenarios. Despite showing results that handle large crowds in real time, their approach does not allow agents to come into direct contact with their neighbours due to the cell size limitation. Recently, video game developers have started to use middleware physics engines [13] to simulate in-game character and object motion. A physics-middleware-based crowd simulation was implemented in [19]. Despite being able to handle large crowds at interactive frame-rate, the behaviour of agents was deemed to be too simplistic as no collision avoidance of static or dynamic entities were incorporated.

Our approach of crowd simulation combines a modified layered AI approach consisting of a physics middleware engine and Helbing's Social Forces. We extend and improve the layered AI approach and handle agent navigation using a physics middleware engine driven by the social force approach.

### **2.2 The uniqueness of airports**

Airport terminals have attracted the attention of a broad range of scientists who have tried to capture and summarise the uniqueness of airport user experiences [3, 9, 10, 14, 16]. Commonly identified traits of airports are people meeting events, traveller purposes, time pressures, high levels of intrusive surveillance through security monitoring, and heightened user emotions. Each of those traits affect airport users in their behaviour and experiences, which can be said to be unique to airports.

Lee et al. [9] try to automatically track meeting events with pedestrian detection and tracking methods using airport surveillance video, due to the unique frequency

and density of several different types of people meeting events. These meeting events can be, for instance, people coming towards each other, people waiting for others, a person joining an existing group, etc. They present a people trajectory tracking and meeting event method and apply it on selected video source data from London Gatwick Airport. Their results show the precision and strengths of their method, however they identified event recognition issues, such as false alarms.

Bowes [3] investigates the impact of emotion and time on the shopping behaviour of people in an international airport. The main purpose of the research was to determine if there is a relationship between an individual's emotional state as influenced by the airport terminal environment and the desire to visit an airport retail store. In their results, airport terminals retailers were compared to high-street retailers. Preliminary results identified key points which contribute to the unique nature of airports:

- *Purpose*: Shopping is not the primary purpose for visiting an airport terminal. The main purpose is to travel or meet/greet those who are travelling.
- *Time pressures*: Flight schedules require airplanes to be leaving and landing on time.
- *Heightened emotions*: Airport events (landing/departure of an airplane), time pressure and travelling purposes lead to potentially charged emotional events.

Other researchers using airport environments in their studies [10, 14, 16] also commonly noted that airports generate heightened emotions amongst their users for similar reasons as discussed above. They also examine additional qualities intrinsic to airports such as the constantly high level of security and safety measures, which result in airports being one of the most authoritarian structures available to the free use of civilians. Whilst the main purpose of the security and safety measures is to ensure a safe journey, people's privacies and emotions are altered through the various security checkpoints, the constant surveillance, the frequent safety announcements and the extra power of authorities to search and detain individuals. Such safety traits are unique to the context of airports and impact heavily on the behaviour and emotional responses of airport users.

Recently, video game research have started to exploit such unique features of airport terminals. Linehan et al. [10] developed a pervasive mobile phone game named 'BlowTooth' as a tool in order to identify whether airport safety and security measures can be used to create a novel gaming experience. A passage in the video game 'Call of Duty: Modern Warfare 2' [4] engages the player in a terrorist attack on an international airport.

Although being novel ways to use airports in entertainment there are several issues with the above approaches involving privacy and ethical implications. For example, the authors of 'BlowTooth' are aware of the edginess and risqué nature of the idea of playing pervasive games in a non-traditional playful or fun environment such as airports. The airport scene in 'Call of Duty: Modern Warfare 2' also generated discussions [7] and has been recently suggested to mirror events caused during the recent Moscow Airport attack [2], and therefore further suggesting that airports might be inappropriate environments for entertainment purposes.

### 3 Framework

As airports contain unique features as discussed in the previous section, we have decided to use an airport terminal, in this case the London Gatwick South Terminal [11] as a backdrop for our crowd simulation environment. To our knowledge, no existing crowd simulation has specifically exploited the uniqueness of airport traveller experience as a backdrop for a real-time virtual crowd setting in video game situations. In this context, we set out to design and implement a video game environment that combines the middleware PhysX [13] and a scalable layered intelligence technique [21] from the video game industry, using Dijkstra’s algorithm [5] to pre-compute navigation information, with social-forces [6] driven agents as characters in order to reproduce some unique and interesting airport scenarios, which would in return enrich the feelings of realism and believability for the player. The decision to use an airport scenario was chosen to demonstrate and harness its unique features and to address the lack of video game related research on real-time crowd simulation inside an airport environment.

#### 3.1 Static and Dynamic Representation

We take as our point of departure the results generated by our previous work on virtual crowds, which simulated a large number of real-time agents in an urban context [19]. The results of the previous work demonstrate that a combination of PhysX to simulate the agents’ reactive behaviour and Millington’s goal oriented algorithm [12] to produce deliberative reasoning can generate real-time large crowds at interactive frame-rate ( $>30$  frames-per-second).

In this work, we decided to use a similar hybrid architecture for simulating in real-time a large number of agents in an airport environment. We decided to combine ‘high-level’ deliberative reasoning (using a spatial database approach to store environmental and sociological data) and ‘low-level’ reactive control (using PhysX) to generate sophisticated agent behaviour. Although we only focus on basic navigational behaviour in this paper, we show that it is feasible to use our framework to populate a crowded airport environment.

The concept of spatial databases has been popularised by the video game industry [21]. The basic idea is to store any static or dynamic AI related information in several layers and have agents take decisions based on one or a combination of these layers. Data is separated into different layers, for instance, all navigation related data is stored in one layer. One major advantage of the spatial database approach is its scalability as further layers can be easily added. Banerjee et al. [1] recently extended the spatial database to simulate an agent-based crowd simulation.

We consider our environment as a 2-D surface, which is divided into cells. In contrast to the approach in [1], we do not limit ourselves to a maximum of one agent per cell. Each cell has a sufficient area to contain at least one person of average size and the total amount of individuals it can contain is limited to the cell’s actual physical size. Due to the physical constraints simulated by PhysX, agents placed on the same cell do not overlap. Each cell on the layer contains an average of all the social and environmental data which affect all entities located on the same cell. On top of the grid, we have currently added four independent layers, which contain both

static and dynamic data. In the experiments presented in this paper, we implemented the following layers: *physical layer*, *navigation layer*, *occupancy layer* and *area search layer*:

- *Physical layer* represents information about the environment and all the static obstacles, such as walls, stairs, lifts, info screens, etc.
- *Navigation layer* stores all path-finding data from each cell to each destination cell generated offline by the Dijkstra algorithm.
- *Occupancy layer* represents all information about the occupancy levels of each cell saving information about which individuals are currently occupying which cell.
- *Area search layer* contains information about an unblocked line-of-sight to all the cells within a limited viewing arc in front of each agent's point of view.

We view movement of agents on the 2-D surface as a sum of vectors of all separate weighted forces reflecting the different environmental or sociological influences. The method that we use is best described through the ‘social-forces’ model popularised by [6], which assumes that an agent's movements in a crowd is based on physical and sociological forces, which in return stimulate behaviour. Essentially, our work combines Helbing's Social Forces Model [6] with the spatial database approach as presented by [21]. We define each cell in each layer to contain relevant environmental or sociological data which is summarised into a force vector. This force is applied onto the agent to consequently generate deliberative behaviour.

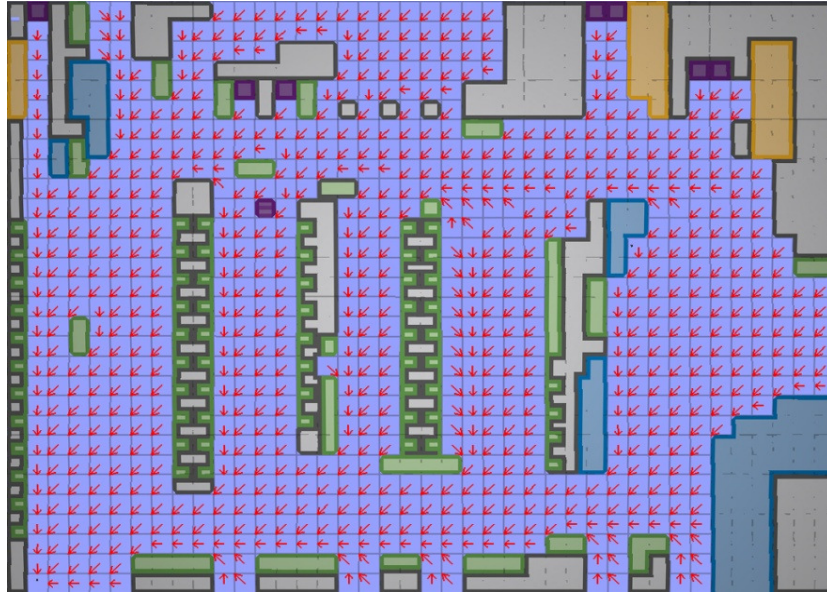
For instance, in our work we define each agent  $A$  from a total of  $N$  agents ( $A = A_1..A_N$ ) to be positioned in the virtual world  $W$ .  $W$  contains a total of  $M$  layers  $L$ , where  $L=L_1..L_M$ .  $Layer_l(x,y)$  represents the value of cell  $(x,y)$  in layer  $l$ . An agent needs to look up all the values of its current cell in each layer, weight them using  $w$  and summarise them as a force  $F$  which is consequently applied onto the agent. The force acting on each agent at each update cycle is given by the dynamic equation:

$$F_A = \sum_{l=0}^M w_l layer_l(x,y). \quad (1)$$

If (1) is applied, the agents try to avoid static obstacles defined in the *physical layer*, while moving towards a goal cell using a pre-calculated path stored in the *navigation layer*. Simultaneously, agents try to keep a velocity-dependent distance from heavily occupied cells, identified through the *occupancy layer*, which lie within an agent's *area search layer*.

### 3.2 Implementation

We implemented the static and dynamic representation discussed in the previous section in Visual Studio 2008 (C++, GLUT, PhysX 2.8.4). We also implemented a simple graphical interface for evaluation purposes. The world  $W$  is defined as  $200 \times 150\text{m}$  in size and sub-divided into cells of  $5 \times 5\text{m}$ . We define the *physical layer* to a similar layout as the London Gatwick South Terminal as shown in Fig. 1. The *physical layer* is used for PhysX static object generation, off-line path generation (Dijkstra's algorithm adapted from [5]) and layout rendering. The *navigation layer* is loaded with the pre-computed path information. Fig. 2 shows an example of flow-field information stored in the *navigational layer*. The data stored in the *occupancy* and *area search* layers are dynamically updated each time an agent moves to one of its neighbouring cells. Agents  $A$  were created as dynamic PhysX actors with the size of an average person. The experiments were carried out on an Intel Core 2 Quad CPU Q6700@2.66GHz (single threaded), 2048MB Ram, GeForce 8800 GT (256MB). Fig. 3 shows a snapshot of the implementation with 1,500 agents (rendered as white dots). At the current stage of our research, the aim is to study the feasibility and scalability of our approach; therefore we recorded the frames-per-second (FPS) with population size varying between 0 and 10,000. Fig. 4 shows the recorded performance of the implementation without graphical visualisation. The FPS readings were obtained after the simulation had run for some time, allowing the average performance to be measured. The results show that the relationship between the number of individuals  $N$  and the performance of the application reduces exponentially with an increasing number of agents and indicates a possible bottleneck of our approach when larger populations are used.



**Fig. 2.** Navigation flow-field produced by Dijkstra's algorithm, stored in the Navigation layer with goal the bottom-left corner.

In most video games, an acceptable minimum interactive frame-rate has been discussed to lie above thirty FPS. The system presented in this paper is able to simulate roughly up to 4,200 agents at interactive frame-rate. Whilst the current actions of agents do not follow *typical* airport behaviour, the current framework can be considered to offer sufficient performance to simulate a large crowd whilst considering future improvements.

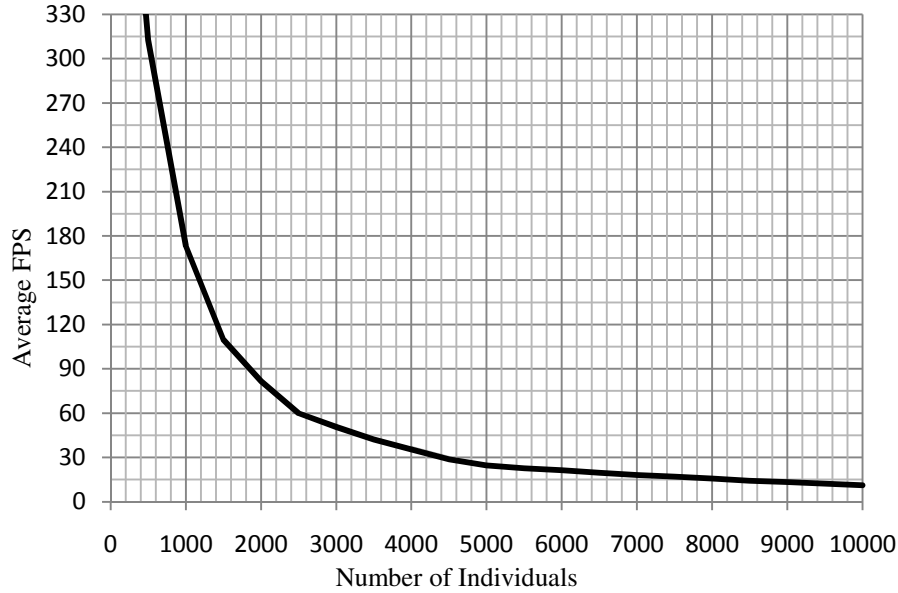


**Fig. 3.** Snapshot of the implementation. Agents are represented as white points, which try to follow the player controlled entity currently situated in the bottom left corner

#### 4 Limitations and Future work

One of the major limitations of our approach is the requirement that data contained in the *navigation layer* has to be pre-calculated offline due to its computationally intensive requirements. This means that changes in the *physical layer*, for instance adding a new static wall section, would require the data in the *navigation layer* to be fully recalculated. Another yet unexplored limitation is the size of each cell. Currently, in our implementation, we are using cell sizes of 5×5m, however, we have not explored the impact of the granularity of the layer data and its resulting effect on the simulation if larger cells sizes are used. Currently the agent's behaviour is limited to following a player controlled entity, thus not reflecting typical airport user behaviour. In future work we plan not only to improve the current limitations but also to extend our layered AI approach to contain more layers which reflect other aspects of typical airport behaviour. We also plan to include group structures to reflect families and other small groups travelling together as a tight, closed unit. We also plan to include situational events, which in return affect an agent's current emotional status, which in return modifies the current behaviour.





**Fig. 4.** Computational performance showing average frames-per-second against number of individuals in the simulation.

## 5 Summary

We have presented a framework that combines a modified spatial database approach with the middleware PhysX in order to simulate a virtual crowd using the social forces approach. We have modified the spatial database approach to contain more than one agent per cell, thus augmenting the spread of sociological and environmental data in each cell. Agent overlapping is prevented by the physical boundaries generated through the middleware PhysX. Agent movement is described through the ‘social-forces’ approach by summarising all the relevant environmental or sociological information spread throughout the cells in each layer. Furthermore, we have applied this framework in an airport terminal environment, which has been discussed to provide a strong backdrop for a crowd simulation due to its uniqueness. We have run several experiments in order to establish the scalability and feasibility of our approach. The framework was shown to produce large crowds at interactive frame-rate. Although it was said that the current agent’s behaviour did not follow a *typical* airport terminal behaviour as influenced by the unique traits of airports, the assessment of the framework’s computational performances and extendibility was deemed to offer sufficient qualities, which in future work will be utilised to implement a crowd simulation that fully harnesses the uniqueness of airport terminals.

## 6 References

1. Banerjee, B., Abukmail, A., and Kraemer, L. Advancing the Layered Approach to Agent-Based Crowd Simulation. Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation (2008)
2. BBC News: Moscow bombing: Carnage at Domodedovo airport. <http://www.bbc.co.uk/news/world-europe-12268662> (2010)
3. Bowes, B. The effects of emotion and time to shop on shopping behaviour in an international airport terminal. presentation to the Association of Consumer Research. pp. 207--214. (2002)
4. Call of Duty Modern Warfare 2, Infinity Ward Studios, Activision. <http://modernwarfare2.infinityward.com/agegate.php> (2009)
5. Dijkstra, E. W. "A note on two problems in connexion with graphs". In: *Numerische Mathematik* 1. pp. 269--271. <http://www-m3.ma.tum.de/twiki/pub/MN0506/WebHome/dijkstra.pdf> (1959)
6. Helbing D, Molnar P., Farkas I. J, and Bolay K. Self-organizing pedestrian movement. *Environment and Planning B* 28. pp. 361--384. (2001)
7. Kotaku: Russian TV links Modern Warfare 2 to Russian Massacre. <http://kotaku.com/5742990/russian-tv-links-modern-warfare-2-to-russian-massacre?skyline=true&s=i> (2010)
8. LeBon, G. *Psychologie des Foules*. Alcan, Paris. (1895)
9. Lee, S., Huang, C. & Nevatia, R. Definition, Detection, and Evaluation of Meeting Events in Airport Surveillance Videos. USC at TRECVID. <http://www-nlpir.nist.gov/projects/tvpubs/tv8.papers/usc.pdf> (2008)
10. Linehan, C. Kirman, B. Lawson, S. Doughty, M. Blowtooth: pervasive gaming in unique and challenging environments. Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems. pp. 2695--2704. (2010)
11. London Gatwick Airport South Terminal Map. [http://www.gatwickairport.com/Documents/Maps/Gatwick\\_South\\_08\\_10.pdf](http://www.gatwickairport.com/Documents/Maps/Gatwick_South_08_10.pdf) (2010)
12. Millington I., *Artificial Intelligence for Games (The Morgan Kaufmann Series in Interactive 3D Technology)*, Morgan Kaufmann Publishers Inc., San Francisco, CA. (2006)
13. Nvidia PhysX. [http://www.nvidia.com/object/physx\\_new.html](http://www.nvidia.com/object/physx_new.html) (2010)
14. Omar, O. & Kent, A. International airport influences on impulsive shopping: trait and normative approach. *International Journal of Retail & Distribution Management*, 29(5). pp. 226--235. (2001)
15. Pelechano, N., Allbeck, J. & Badler, Controlling individual agents in high-density crowd simulation. In Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association. pp. 108--118. (2007)
16. Pepe, A. Reinventing airspace: spectatorship, fluidity, intimacy at PEK T3 ACE: *Journal of Architecture, City & Environment*. 4.10. pp. 9--19. (2009)
17. Reynolds, C. Flocks, herds and schools: A distributed behavioral model. In Proceedings 14th annual conference on Computer graphics and interactive techniques, pp. 25--34. (1987)
18. Reynolds, C. Big fast crowds on PS3. In Proceedings of the 2006 ACM SIGGRAPH Symposium on Videogames. (2006)
19. Szymanczyk, O. & Cielniak, G., 2010. Emotion Modelling And Middleware Use For Group Sized Crowds In Videogames. In: AISB 2010 Symposium: AI and Games. (2010)
20. Thalmann D. and Musse S. R., *Crowd Simulation*, 1st ed. Springer. (2007)
21. Tozour, P. Using Spatial Database for Runtime Spatial Analysis. In: *AI Game Programming Wisdom Volume 2*, pp. 381--390. Charles River Media. (2004)